

VISVESVARAYA TECHNOLOGICAL UNIVERSITY
“JnanaSangama”, Belgaum -590014, Karnataka.



LAB REPORT
on

Object Oriented Java Programming
(23CS3PCOOJ)

Submitted by

Archie Jain K M
(1BM23CS049)

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING



B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)

BENGALURU-560019
Sep-2024 to Jan-2025

**B.M.S. College of Engineering,
Bull Temple Road, Bangalore 560019**
(Affiliated To Visvesvaraya Technological University, Belgaum)
Department of Computer Science and Engineering



CERTIFICATE

This is to certify that the Lab work entitled “Object Oriented Java Programming (23CS3PCOOJ)” carried out by **Archie Jain K M (1BM23CS049)**, who is bonafide student of **B.M.S. College of Engineering**. It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements in respect of an Object Oriented Java Programming (23CS3PCOOJ) work prescribed for the said degree.

Dr .Prasad G R Professor Department of CSE, BMSCE	Dr. Kavitha Sooda Professor & HOD Department of CSE, BMSCE
---	--

Index

Sl. No.	Date	Experiment Title	Page No.
1	09.10.24	Implement Quadratic Equation	4
2	16.10.24	Calculate SGPA of a Student using Class	8
3	23.10.24	Create N book objects	14
4	23.10.24	Create a abstract class and Print the area of shapes	19
5	30.10.24	Create a class Bank	22
6	13.11.24	Packages	31
7	20.11.24	Exception Handling	38
8	27.11.24	Threads	42
9	27.11.24	User interface to perform integer divisions	46
10	27.11.24	Interprocess Communication and deadlock	49

Github Link:

<https://github.com/ArchieJain120/JAVA-LAB>

Program 1

Develop a Java program that prints all real solutions to the quadratic equation $ax^2+bx+c = 0$. Read in a , b , c and use the quadratic formula. If the discriminant $b^2 - 4ac$ is negative, display a message stating that there are no real solutions.

i) Java program to find the roots of the quadratic equation $ax^2+bx+c = 0$. Read in a , b , c and use the quadratic formula. If the discriminant $b^2 - 4ac$ is negative, display a message stating that there are no real solutions.

```
import java.lang.Math;
import java.util.Scanner;
public class Quadratic {
    public static void main (String args[])
    {
        int a, b, c;
        double det, root1, root2;
        Scanner myobj = new Scanner (System.in);
        System.out.println ("Enter the values of a, b, c : ");
        a = myobj.nextInt();
        b = myobj.nextInt();
        c = myobj.nextInt();
        if (a == 0)
            System.out.println ("It is not a quadratic equation");
        else {
            det = b * b - 4 * a * c;
            if (det > 0)
                root1 = ((-b) + (Math.sqrt(det))) / (2 * a);
                root2 = ((-b) - (Math.sqrt(det))) / (2 * a);
                System.out.println ("The equation has two real roots : " + root1 + " and " + root2);
            else if (det == 0)
        }
    }
}
```

```

        root1 = -b / (c * a);
        System.out.println ("The equation has real and
        equal roots");
        System.out.println ("The roots are " + root1);
    } else {
        root1 = (-b) / (c * a);
        root2 = Math.sqrt (-d) / (c * a);
        System.out.println ("The equation has imaginary
        roots");
        System.out.println ("The roots are " + root1 + " and"
        + root2);
    }
}

Output:
Enter the values of a, b, c: 2 5 2
The equation has real and equal roots
The roots are -2.0

```

The equation has real and distinct roots
 The roots are -0.50 and -2.00

Code:

```

import java.util.Scanner;
import java.lang.Math;
class quadratic

```

```

{
    public static void main(String args[])
{
    int a,b,c;
    double det,root,root1,root2;
    Scanner myobj=new Scanner(System.in);
System.out.println("my name is archie");
System.out.println("my usn is 1bm23cs049");
    System.out.println("Enter the values of a,b,c:");
    a=myobj.nextInt();
    b=myobj.nextInt();
    c=myobj.nextInt();

    if(a==0)
    {
        System.out.println("It is not a quadratic equation");
    }
    else
    {
        det=(b*b)-(4*a*c);
        if(det>0)
        {
            root1=((( -b)+(Math.sqrt(det)))/(2*a));
            root2=((( -b)-(Math.sqrt(det)))/(2*a));
            System.out.println("It equation has real and distinct roots");
            System.out.println("The roots are"+root1+"and"+root2);
        }
        else if(det==0)
        {
            root=-b/(2*a);
            System.out.println("It equation has real and equal roots");
            System.out.println("The roots are"+root);
        }
        else
        {
            root1=(-b)/(2*a);
            root2=Math.sqrt(-det)/(2*a);
            System.out.println("It equation has imaginary roots");
            System.out.println("The roots are"+root1+"and"+root2);
        }
    }
}
}

```

Output:

```
C:\Users\bmsce\Desktop\49>javac quadratic.java

C:\Users\bmsce\Desktop\49>java quadratic
my name is archie
my usn is 1bm23cs049
Enter the values of a,b,c:
1 4 4
It equation has real and equal roots
The roots are-2.0

C:\Users\bmsce\Desktop\49>java quadratic
my name is archie
my usn is 1bm23cs049
Enter the values of a,b,c:
2 5 2
It equation has real and distinct roots
The roots are-0.5 and -2.0

C:\Users\bmsce\Desktop\49>java quadratic
my name is archie
my usn is 1bm23cs049
Enter the values of a,b,c:
1 -3 2
It equation has real and distinct roots
The roots are 2.0 and 1.0

C:\Users\bmsce\Desktop\49>java quadratic
my name is archie
my usn is 1bm23cs049
Enter the values of a,b,c:
0 3 5
It is not a quadratic equation

C:\Users\bmsce\Desktop\49>java quadratic
my name is archie
my usn is 1bm23cs049
Enter the values of a,b,c:
4 6 1
It equation has real and distinct roots
The roots are -0.19098300562505255 and -1.3090169943749475

C:\Users\bmsce\Desktop\49>java quadratic 1 1 1
my name is archie
my usn is 1bm23cs049
Enter the values of a,b,c:
1 1 1
It equation has imaginary roots
The roots are 0.0 and 0.8660254037844386
```

Program 2

Develop a Java program to create a class Student with members usn, name, an array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student.

2) Develop a Java program to create a class student with members usn, name, an array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student.

```
import java.util.Scanner;
class student
{
    String usn;
    String name;
    int [ ] credits;
    int [ ] marks;
    void accept_details()
    {
        Scanner mark = new Scanner(System.in);
        System.out.println("Enter USN:");
        USN = mark.nextLine();
        System.out.println("Enter name:");
        Name = mark.nextLine();
        System.out.println("Enter number of subjects:");
        int n = mark.nextInt();
        credits = new int[n];
        marks = new int[n];
        for (int i=1; i<n; i++)
        {
            System.out.println("Enter credit and marks");
            credits[i] = mark.nextInt();
            marks[i] = mark.nextInt();
        }
        void sgpacal()
    }
}

double sgpa;
int cred=0, mark=0;
for (cred=0, mark=0;
     for (int i=0; i<n; i++)
     {
         cred += credits[i];
         mark += marks[i];
     }
     System.out.println("Total credits:" + cred);
     System.out.println("Total marks:" + mark);
     for (int i=0; i<n; i++)
     {
         double sgpa = ((marks[i]/10.0)) * credits[i];
         System.out.println("SGPA is :" + sgpa);
     }
     void displaydetails()
     {
         System.out.println("Name:" + mark.name);
         System.out.println("USN:" + mark.usn);
         for (int i=1; i<n; i++)
         {
             System.out.println("credit for" + i + credits[i]);
             System.out.println("marks for" + i + marks[i]);
         }
         public static void main(String [ ] args)
         {
             student obj = new Student();
             obj.acceptdetails();
             obj.displaydetails();
             obj.sgpacal();
         }
    }
}
```

Output

Enter details of student 1
My name is Riya
Enter name: riyaan
Enter usn : ICS3476
Enter marks for subject : 1
87
Enter credits for subject : 1
4
Enter marks for subject : 2
83
Enter credits for subject : 2
4
Enter marks for subject : 3
87
Enter credits for subject : 3
3
Enter marks for subject 4
78
Enter credits for subject 4
3
Enter marks for subject 5
85
Enter credits for subject 5
3
Enter marks for subject 6
90
Enter credits for subject 6
1
Enter marks for subject 7
93
Enter credits for subject 7
1

Enter marks for subject : 8
92
Enter credits for subject : 8
1

Student - 1

Name riya
USN ICS345
SGPA : 9.6

Code:

```
import java.util.*;

class Stud_det {
    int m[] = new int[8];
    int c[] = new int[8];
    int p[] = new int[8];
    int g, sum;
    String name, usn;
    double sgpa;
    Scanner s = new Scanner(System.in);
    void getdetails()
    {
        System.out.println("My name is archie");
        System.out.println("My usn is 1BM23CS049");
        System.out.println("Enter name:");
        name = s.next();
        System.out.println("Enter usn:");
        usn = s.next();
        for (int i = 0; i < 8; i++) {
            System.out.println("Enter marks of subject:"+ (i+1));
            m[i] = s.nextInt();
            System.out.println("Enter credits for subject:" + (i+1));
            c[i] = s.nextInt();
        }
    }

    void gradepoint() {
        for (int i = 0; i < 8; i++) {
            if (m[i] >= 90 && m[i] <= 100)
                p[i] = 10;
            else if (m[i] >= 80 && m[i] < 90)
                p[i] = 9;
            else if (m[i] >= 70 && m[i] < 80)
                p[i] = 8;
            else if (m[i] >= 60 && m[i] < 70)
                p[i] = 7;
            else if (m[i] >= 50 && m[i] < 60)
                p[i] = 6;
            else if (m[i] >= 40 && m[i] < 50)
                p[i] = 5;
            else
                p[i] = 0;
        }
    }
}
```

```

    }

    void calculate() {
        for (int i = 0; i < 8; i++) {
            g += c[i] * p[i];
        }
        for (int i = 0; i < 8; i++) {
            sum += c[i];
        }
        sgpa = g / sum;
    }

    void display() {
        System.out.println("Name:" + name);
        System.out.println("USN:" + usn);
        System.out.println("SGPA:" + sgpa);
    }
}

class student {
    public static void main(String a[]){
        Stud_det s1[]=new Stud_det[3];
        for(int i=0;i<3;i++){
        {
            s1[i]=new Stud_det();
        }
        for(int i=0;i<3;i++){
        {
            System.out.println("Enter details of student:"+ (i+1));
            s1[i].getdetails();
        }
        for(int i=0;i<3;i++)
        {
            s1[i].gradepoint();
            s1[i].calculate();
        }
        for(int i=0;i<3;i++)
        {
            System.out.println("Student-"+(i+1));
            s1[i].display();
        }
    }
}

```

Output:

```
C:\Users\bmsce\Desktop\49>javac student.java
C:\Users\bmsce\Desktop\49>java student
Enter details of student:2
My name is archie
My usn is 1BM23CS049
Enter name:
riya
Enter usn:
1cse345
Enter marks of subject:1
89
Enter credits for subject:1
4
Enter marks of subject:2
97
Enter credits for subject:2
4
Enter marks of subject:3
87
Enter credits for subject:3
3
Enter marks of subject:4
83
Enter credits for subject:4
3
Enter marks of subject:5
90
Enter credits for subject:5
3
Enter marks of subject:6
90
Enter credits for subject:6
1
Enter marks of subject:7
96
Enter credits for subject:7
1
Enter marks of subject:8
93
Enter credits for subject:8
1
Enter details of student:2
My name is archie
My usn is 1BM23CS049
Enter name:
rihaan
Enter usn:
1cs3476
Enter marks of subject:1
87
Enter credits for subject:1
4
Enter marks of subject:2
83
Enter credits for subject:2
4
Enter marks of subject:3
87
Enter credits for subject:3
3
Enter marks of subject:4
78
Enter credits for subject:4
3
Enter marks of subject:5
85
Enter credits for subject:5
3
Enter marks of subject:6
90
Enter credits for subject:6
1
Enter marks of subject:7
93
Enter credits for subject:7
1
Enter marks of subject:8
92
Enter credits for subject:8
1
```

```
Enter credits for subject:8
1
Enter details of student:3
My name is archie
My usn is 1BM23CS049
Enter name:
reema
Enter usn:
1be3487
Enter marks of subject:1
88
Enter credits for subject:1
4
Enter marks of subject:2
85
Enter credits for subject:2
4
Enter marks of subject:3
89
Enter credits for subject:3
3
Enter marks of subject:4
80
Enter credits for subject:4
3
Enter marks of subject:5
78
Enter credits for subject:5
3
Enter marks of subject:6
89
Enter credits for subject:6
1
Enter marks of subject:7
87
Enter credits for subject:7
1
Enter marks of subject:8
90
Enter credits for subject:8
1
Student-1
Name:riya
USN:1cse345
SGPA=:9.0
Student-2
Name:rihaan
USN:1cs3476
SGPA=:9.0
Student-3
Name:reema
USN:1be3487
SGPA=:8.0
```

Program 3

Create a class Book which contains four members: name, author, price, num_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a `toString()` method that could display the complete details of the book. Develop a Java program to create n book objects.

SURYA Gold Date _____ Page _____	SURYA Gold Date _____ Page _____
<p>3) Create a class Book which contains four members: name, author, price, num_of_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a <code>toString()</code> method that could display the complete details of the book. Develop a Java program to create n book objects.</p> <pre>import java.util.Scanner; class Book { private String name; private String Author; private int price; private int numpages; public Book(String name, String author, int price, int numpages) { this.name = name; this.author = author; this.price = price; this.numpages = numpages; } public String getName() { return name; } public String getAuthor() { return author; } public int getPrice() { return price; } public int getNumpages() { return numPages; } public void SetName(String name) { this.name = name; } public void SetAuthor(String author) { this.author = author; } public void setPrice(int price) { this.price = price; } public void setNumpages(int numPages) { this.numpages = numPages; } public String toString() { return "Book Details: \n" + "Name: " + name + "\n" + "Author: " + author + "\n" + "Price: " + price + "\n" + "Number of pages: " + numPages + "\n"; } }</pre>	<p>3) Create a class Book which contains four members: name, author, price, num_of_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a <code>toString()</code> method that could display the complete details of the book. Develop a Java program to create n book objects.</p> <pre>public int getPrice() { return price; } public int getNumpages() { return numPages; } public void SetName(String name) { this.name = name; } public void SetAuthor(String author) { this.author = author; } public void setPrice(int price) { this.price = price; } public void setNumpages(int numPages) { this.numpages = numPages; } public String toString() { return "Book Details: \n" + "Name: " + name + "\n" + "Author: " + author + "\n" + "Price: " + price + "\n" + "Number of pages: " + numPages + "\n"; }</pre>

SURYA Gold
Date _____ Page _____

```

public class Main
{
    public static void main ( String [] args )
    {
        Scanner sc = new Scanner ( System.in );
        System.out.println ( " Enter the number of books : " );
        int n = sc.nextInt ();
        sc.nextLine ();
        Book [] books = new Book [n];
        for ( int i=0; i<n; i++ )
        {
            System.out.println ( " Book Details : " );
            System.out.println ( " Book name : " );
            String name = sc.nextLine ();
            System.out.println ( " Author's name : " );
            String author = sc.nextLine ();
            System.out.println ( " Book's price : " );
            int price = sc.nextInt ();
            System.out.println ( " Number of pages : " );
            int numPages = sc.nextInt ();
            books[i] = new Book ( name, author, price, numPages );
        }
        System.out.println ( " Details of all books : " );
        for ( Book book : books )
        {
            System.out.println ( book );
        }
    }
}

```

Output

Enter the number of books : 3

Book details :

Book name :

~~Once~~ I came upon a light house

Author's name : Shantanu Naidu

Book's price : 200

Number of pages : 175

Book details :

Book name : Most and More

Author's name : Mahatma Ra

Book's price : 275

Number of pages : 250

Book details :

Book name : Two states

Author's name : Chetan Bhagat

Book's price : 175

Number of pages : 200

Details of all books :

Book@f18ufc6

Book@f18ufc7

Book@f18ufc8

Code:

```
import java.util.Scanner;
class Book
{
    private String name;
    private String author;
    private int price;
    private int numpages;

    public Book(String name, String author, int price, int numpages)
    {
        this.name=name;
        this.author=author;
        this.price=price;
        this.numpages=numpages;
    }
    public String getname()
    {
        return name;
    }
    public String getauthor()
    {
        return author;
    }
    public int getprice()
    {
        return price;
    }
    public int getnumpages()
    {
        return numpages;
    }
    public void setname(String name)
    {
        this.name=name;
    }
    public void setauthor(String author)
    {
        this.author=author;
    }
    public void setprice(int price)
```

```

{
    this.price=price;
}
public void setnumpages(int numpages)
{
this.numpages=numpages;
}
public String tostring()
{
    String name, author, price, numPages;
    name = "Book name: " + this.name + "\n";
    author = "Author name: " + this.author + "\n";
    price = "Price: " + this.price + "\n";
    numPages = "Number of pages: " + this.numpages + "\n";
    return name + author + price + numpages;
}
}
public class bookse
{
public static void main(String []args)
{
Scanner sc=new Scanner(System.in);
System.out.println("Enter the number of books:");
int n=sc.nextInt();
sc.nextLine();
Book[] books=new Book[n];
for(int i=0;i<n; i++)
{
System.out.println("Book details:");
System.out.println("Book name:");
String name=sc.nextLine();
System.out.println("Authors name:");
String author=sc.nextLine();
System.out.println("Book price:");
int price=sc.nextInt();
System.out.println("Number of pages:");
int numpages=sc.nextInt();
books[i]=new Book (name,author,price,numpages);
}
System.out.println("Details of all books:");
for(Book book:books)
{
    System.out.println(book);
}}}

```

Output:

```
C:\Users\HP\Documents>javac bookse.java
```

```
C:\Users\HP\Documents>java bookse
```

```
My name is archie jain
```

```
USN:1BM23CS049
```

```
Enter the number of books:
```

```
3
```

```
Book details:
```

```
Book name:
```

```
once i came upon a light house
```

```
Authors name:
```

```
shantanu naidu
```

```
Book price:
```

```
275
```

```
Number of pages:
```

```
250
```

```
Book details:
```

```
Book name:
```

```
Authors name:
```

```
most and more ,mahatria ra
```

```
Book price:
```

```
200
```

```
Number of pages:
```

```
175
```

```
Book details:
```

```
Book name:
```

```
Authors name:
```

```
two states , chetan bhagat
```

```
Book price:
```

```
170
```

```
Number of pages:
```

```
170
```

```
Details of all books:
```

```
Book@5f184fc6
```

```
Book@3feba861
```

```
Book@5b480cf9
```

Program 4

Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea() that prints the area of the given shape.

SURYA Gold
Date _____ Page _____

4) Develop a Java program to create an abstract class named shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and Circle such that each one of their classes contain only the method printArea() that prints the area of the given shape.

```
import java.util.Scanner;
abstract class Shape {
    int dimension1;
    int dimension2;
    abstract void printArea();
}

class Rectangle extends Shape {
    Rectangle(int length, int breadth) {
        this.dimension1 = length;
        this.dimension2 = breadth;
    }

    void printArea() {
        double area = dimension1 * dimension2;
        System.out.println("Area of Rectangle: " + area);
    }
}

class Triangle extends Shape {
    Triangle(int base, int height) {
        this.dimension1 = base;
        this.dimension2 = height;
    }

    void printArea() {
        double area = 0.5 * dimension1 * dimension2;
        System.out.println("Area of Triangle: " + area);
    }
}
```

```

Date _____ Page _____
this. dimension2 = height;
}
{
    void printArea() {
        double area = 0.5 * dimension1 * dimension2;
        System.out.println("Area of triangle = " + area);
    }
}

class Circle extends Shape {
    {
        Circle (int radius) {
            this. dimension1 = radius;
            this. dimension2 = 0;
        }

        void printArea() {
            double area = Math.PI * dimension1 * dimension1;
            System.out.println("Area of circle = " + area);
        }
    }

    public class Shapes {
        public static void main (String [] args) {
            Scanner sc = new Scanner (System.in);
            System.out.println ("Enter the length and breadth of the rectangle:");
            int length = sc.nextInt ();
            int breadth = sc.nextInt ();
            Shape rectangle = new Rectangle (length, breadth);
            rectangle.printArea();
        }
    }
}

```

```
System.out.println ("Enter the base and height of the
triangle :");
```

```
int base = sc.nextInt();
```

```
int height = sc.nextInt();
```

```
Shape triangle = new Triangle (base, height);
```

```
triangle.printArea();
```

```
System.out.println ("Enter the radius of the circle :");
```

```
int radius = sc.nextInt();
```

```
Shape circle = new Circle (radius);
```

```
circle.printArea();
```

```
}
```

```
}
```

Output

Enter the length and breadth of the rectangle: 12 10

Area of Rectangle: 120.0

Enter the base and height of another triangle: 5 6

Area of triangle: 15.0

Enter the radius of the circle: 7

Area of circle: 153.93804

Code:

```
import java.util.Scanner;
abstract class Shape
{
    int dimension1;
    int dimension2;
    abstract void printArea();
}
class Rectangle extends Shape
{
    Rectangle(int length, int breadth)
```

```

    {
        this.dimension1 = length;
        this.dimension2 = breadth;
    }
    void printArea()
    {
        double area = dimension1 * dimension2;
        System.out.println("Area of Rectangle: " + area);
    }
}
class Triangle extends Shape
{
    Triangle(int base, int height)
    {
        this.dimension1 = base;
        this.dimension2 = height;
    }

    void printArea()
    {
        double area = 0.5 * dimension1 * dimension2;
        System.out.println("Area of Triangle: " + area);
    }
}

class Circle extends Shape
{
    Circle(int radius)
    {
        this.dimension1 = radius;
        this.dimension2 = 0;
    }

    void printArea()
    {
        double area = Math.PI * dimension1 * dimension1;
        System.out.println("Area of Circle: " + area);
    }
}

public class shapes
{
    public static void main(String[] args)

```

```

{
    Scanner sc = new Scanner(System.in);

    System.out.print("Enter the length and breadth of the rectangle: ");
    int length = sc.nextInt();
    int breadth = sc.nextInt();
    Shape rectangle = new Rectangle(length, breadth);
    rectangle.printArea();

    System.out.print("Enter the base and height of the triangle: ");
    int base = sc.nextInt();
    int height = sc.nextInt();
    Shape triangle = new Triangle(base, height);
    triangle.printArea();

    System.out.print("Enter the radius of the circle: ");
    int radius = sc.nextInt();
    Shape circle = new Circle(radius);
    circle.printArea();
}
}

```

Output:

```

C:\Users\HP\Documents>javac Area.java
C:\Users\HP\Documents>javac Area.java
C:\Users\HP\Documents>java Area
Name:Archie Jain
USN:1BM23CS049
Enter the length and breadth of the rectangle
10
15
Area of Rectangle: 150
Enter the base and height of the triangle
12
6
Area of Triangle: 36.0
Enter the radius of the circle
6
Area of Circle: 113.09733552923255
Area of Rectangle: 150
Area of Triangle: 36.0
Area of Circle: 113.09733552923255

```

Program 5

Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed.

Create a class Account that stores customer name, account number and type of account. From this derive the classes Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

- a) Accept deposit from customer and update the balance.
- b) Display the balance.
- c) Compute and deposit interest
- d) Permit withdrawal and update the balance. Check for the minimum balance, impose penalty if necessary and update the balance

SURYA Gold
Date _____ Page _____

Bank Account.

```

5) import java.util.Scanner;
class PrintInfo {
    static void print() {
        System.out.println("Name: " + name);
        System.out.println("Account Number: " + accountNumber);
    }
}
abstract class Account {
    String customerName;
    String accountType;
    String accountNumber;
    double balance;
    public Account(String customerName, String accountType, String accountNumber) {
        this.customerName = customerName;
        this.accountType = accountType;
        this.accountNumber = accountNumber;
        this.balance = 0.0;
    }
    public void deposit(double amount) {
        balance += amount;
        System.out.println("Deposited amount is: " + amount);
        displayBalance();
    }
    public void displayBalance() {
        System.out.println("Current balance is: " + balance);
    }
    public abstract void withdraw(double amount);
}

```

SURYA Gold
Date _____ Page _____

```

class Saveacct extends Account {
    double interestRate;
    public Saveacct(String customerName, String accountNumber, double interestRate) {
        super(customerName, "Savings", accountNumber);
        this.interestRate = interestRate;
    }
    public void compoundDeposit() {
        double interest = balance * (interestRate / 100);
        deposit(interest);
        System.out.println("Interest of " + interest +
                           " deposited");
    }
    public void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawn amount is: " +
                               amount);
        } else {
            System.out.println("Insufficient amount for
                               withdrawal.");
            return;
        }
        displayBalance();
    }
}

```

SURYA Gold
Date _____ Page _____

```

class CurAcct extends Account {
    private static final double minBalance = 1000.0;
    private static final double servicecharge = 50.0;

    public CurAcct (String customerName, String accountNumber) {
        super (customerName, "current", accountNumber);
    }

    public void withdraw (double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println ("Withdrawn amount is: " + amount);
        } else {
            System.out.println ("Insufficient amount for withdrawal");
        }
        return;
    }

    if (balance < minBalance) {
        balance -= servicecharge;
        System.out.println ("Minimum balance not maintained");
        System.out.println ("Service charge of: " + servicecharge + " included");
    }

    displayBalance();
}

```

SURYA Gold
Date _____ Page _____

```

public class bank {
    public static void main (String [] args) {
        printInfo.print();
        Scanner scanner = new Scanner (System.in);
        System.out.println ("Enter your account type
                           (savings/ current): ");
        String accountType = scanner.nextLine();
        System.out.println ("Enter account number: ");
        String accountNumber = scanner.nextLine();
        System.out.println ("Enter your name: ");
        String customerName = scanner.nextLine();

        Account account;
        if (accountType.equals ("Savings"))
            account = new SavAcct (customerName, accountNumber,
                                   interestRate);
        else
            account = new CurAcct (customerName, accountNumber);

        while (true) {
            System.out.println ("1. Deposit\n2. Withdraw\n3. Display Balance\n4. Exit");
            int choice = scanner.nextInt();
            switch (choice) {
                case 1:
                    System.out.println ("Enter amount to deposit: ");
                    double depositAmount = scanner.nextDouble();
                    account.deposit (depositAmount);

```

```

break;
case 2: System.out.println("Enter amount to withdraw : ");
double withdrawAmount = scanner.nextDouble();
account.withdraw(withdrawAmount);
break;
case 3: account.displayBalance();
break;
case 4: System.out.println("Exit");
scanner.close();
return;
default: System.out.println("Try Again!");
}
}
}

Output
Enter account type (savings /current): savings
Enter account name: renu
Enter account number: 123098
Enter initial balance and interest rate: 2500 3.5
1. Deposit 2: Display balance 3. Withdraw 4. Compute
Interest 5. Exit
1.
Enter deposit amount: 1200
2.
Savings balance: 37000.0

```

	SURYA Gold	SURYA Gold
	Date _____	Page _____
break;	3. withdraw and display 4. Compute interest	
case 2:	Enter withdrawal amount:	
	1900	
	2. display balance	
	Savings balance: 35100.	

```

import java.util.Scanner;
class PrintInfo {
    static void print() {
        System.out.println("Name: Archie Jain");
        System.out.println("USN: 1BM23CS049");
    }
}

abstract class Account {
    String customerName;
    String accountType;
    String accountNumber;
    double balance;

    public Account(String customerName, String accountType, String accountNumber) {
        this.customerName = customerName;
        this.accountType = accountType;
        this.accountNumber = accountNumber;
        this.balance = 0.0;
    }

    public void deposit(double amount) {
        balance += amount;
        System.out.println("Deposited amount is: " + amount);
        displayBalance();
    }

    public void displayBalance() {
        System.out.println("Current balance is: " + balance);
    }

    public abstract void withdraw(double amount);
}

class SavAcct extends Account {
    double interestRate;

    public SavAcct(String customerName, String accountNumber, double interestRate) {
        super(customerName, "savings", accountNumber);
        this.interestRate = interestRate;
    }
}

```

```

public void compoundDeposit() {
    double interest = balance * (interestRate / 100);
    deposit(interest);
    System.out.println("Interest of " + interest + " deposited");
}

public void withdraw(double amount) {
    if (amount <= balance) {
        balance -= amount;
        System.out.println("Withdrawn amount is: " + amount);
    } else {
        System.out.println("Insufficient amount for withdrawal.");
        return;
    }
    displayBalance();
}
}

class CurAcct extends Account {
    private static final double minBalance = 1000.0;
    private static final double serviceCharge = 50.0;

    public CurAcct(String customerName, String accountNumber) {
        super(customerName, "current", accountNumber);
    }

    public void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawn amount is: " + amount);
        } else {
            System.out.println("Insufficient amount for withdrawal.");
            return;
        }

        if (balance < minBalance) {
            balance -= serviceCharge;
            System.out.println("Minimum balance not maintained");
            System.out.println("Service charge of: " + serviceCharge + " included");
        }
        displayBalance();
    }
}

```

```

public class bank {
    public static void main(String[] args) {
        PrintInfo.print();

        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter your account type (savings/current):");
        String accountType = scanner.nextLine();
        System.out.println("Enter account number:");
        String accountNumber = scanner.nextLine();
        System.out.println("Enter your name:");
        String customerName = scanner.nextLine();

        Account account;
        if (accountType.equals("savings")) {
            System.out.println("Enter the interest rate:");
            double interestRate = scanner.nextDouble();
            account = new SavAcct(customerName, accountNumber, interestRate);
        } else {
            account = new CurAcct(customerName, accountNumber);
        }

        while (true) {
            System.out.println("1. Deposit\n2. Withdraw\n3. Display Balance\n4.
Exit");
            int choice = scanner.nextInt();
            switch (choice) {
                case 1:
                    System.out.println("Enter amount to deposit:");
                    double depositAmount = scanner.nextDouble();
                    account.deposit(depositAmount);
                    break;
                case 2:
                    System.out.println("Enter amount to withdraw:");
                    double withdrawAmount = scanner.nextDouble();
                    account.withdraw(withdrawAmount);
                    break;
                case 3:
                    account.displayBalance();
                    break;
                case 4:
                    System.out.println("Exit");
                    scanner.close();
                    return;
                default:

```

```
        System.out.println("Try again");
    }
}
}
}
```

Output:

```
C:\Users\Admin\Documents>javac bank.java
C:\Users\Admin\Documents>java bank
Name: Archie Jain
USN: 1BM23CS049
Enter your account type (savings/current):
savings
Enter account number:
12456
Enter your name:
arya
Enter the interest rate:
3.5
1. Deposit
2. Withdraw
3. Display Balance
4. Exit
1
Enter amount to deposit:
20000
Deposited amount is: 20000.0
Current balance is: 20000.0
1. Deposit
2. Withdraw
3. Display Balance
4. Exit
2
Enter amount to withdraw:
7500
Withdrawn amount is: 7500.0
Current balance is: 12500.0
1. Deposit
2. Withdraw
3. Display Balance
4. Exit
3
Current balance is: 12500.0
1. Deposit
2. Withdraw
3. Display Balance
4. Exit
4
Exit
```

Program 6

Create a package CIE which has two classes- Student and Internals. The class Personal has members like usn, name, sem. The class Internals has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of Student. This class has an array that stores the SEE marks scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses.

<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%;">Date _____</td><td style="width: 10%;">Page _____</td></tr> </table>		Date _____	Page _____
Date _____	Page _____		
<p>6. Create a package CIE which has two classes- Student and Internals. The student has members like usn, name, sem. The class Internals derived from student has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class external which is derived class of student. This class has an array that stores the SEE marks scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses.</p> <pre> package CIE; import java.util.Scanner; public class Student { protected String usn; protected String name; protected int sem; } public void inputStudentDetails() { Scanner s = new Scanner(System.in); System.out.println("Enter USN:"); this.usn = s.nextLine(); System.out.println("Enter name:"); this.name = s.nextLine(); System.out.println("Enter Semester:"); this.sem = s.nextInt(); } public void displayStudentDetails() { } </pre>	<p>SURYA Gold</p> <p>Date _____ Page _____</p> <pre> System.out.println("USN: " + usn); System.out.println("Name: " + name); System.out.println("Semester: " + sem); } class InternalMarks { protected int marks[] = new int[5]; } public void inputCIEmarks() { Scanner s = new Scanner(System.in); System.out.println("Enter Internal marks for 5 courses:"); for(int i=0; i<5; i++) { System.out.print("Course " + (i+1) + ": "); marks[i] = s.nextInt(); } } public void displayCIEmarks() { System.out.println("Internal Marks:"); for(int i=0; i<5; i++) { System.out.print("Course " + (i+1) + ": " + marks[i]); } } </pre>		
<p>3</p>	<p>3</p>		

SURYA Gold
Date _____ Page _____

```

package SEE;
import CIE.Internals;
import java.util.Scanner;

public class Externals extends Internals {
    protected int externalMarks [] = new int [5];
    protected int finalMarks [] = new int [5];

    public Externals() {
        externalMarks = new int [5];
        finalMarks = new int [5];
    }

    public void inputSEEmarks() {
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter External Marks for 5 Courses:");
        for (int i=0; i<5; i++) {
            System.out.println ("Course " + (i+1) + ":");
            externalMarks [i] = sc.nextInt();
        }
    }

    public void displayFinalMarks() {
        for (int i=0; i<5; i++) {
            finalMarks [i] = marks [i] + externalMarks [i];
        }
    }

    public void displayFinalMarks() {
        displayStudentDetails();
        displayCIEmarks();
    }
}

```

Date _____

```

System.out.println ("External Marks:");
for (int i=0; i<5; i++) {
    System.out.println ("Course " + (i+1) + ":" +
        externalMarks [i]);
}

System.out.println ("Final Marks:");
for (int i=0; i<5; i++) {
    System.out.println ("Course " + (i+1) + ":" +
        finalMarks [i]);
}

import SEE.Externals;
import java.util.Scanner;
public class Main {
    public static void main (String [] args) {
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter number of students:");
        int n = sc.nextInt();
        Externals [] students = new Externals [n];
        for (int i=0; i<n; i++) {
            students [i] = new Externals ();
            students [i] = inputStudentDetails ();
            students [i] = inputCIEmarks ();
            students [i] = inputSEEmarks ();
            students [i] = calculateFinalMarks ();
        }
    }
}

```

```

for (int i=0; i<n; i++) {
    students[i].displayFinalMarks();
}
System.out.println("Total marks = " + total);
}

Enter number of students : 5
Enter USN : 1286759
Enter Name : Siri
Enter Semester : 3
Enter Internal Marks for 5 courses:
Course 1 : 34
Course 2 : 35
Course 3 : 32
Course 4 : 37
Course 5 : 40
Enter External Marks for 5 courses:
Course 1 : 38
Course 2 : 39
Course 3 : 40
Course 4 : 40
Course 5 : 30
USN : 1286759
Name : Siri
Semester : 3
Internal Marks : [34, 35, 32, 37, 40]
course 1 : 34 course 2 : 35 course 3 : 32
course 4 : 37 course 5 : 40
External Marks : [38, 39, 40, 40, 30]
course 1 : 38 course 2 : 39 course 3 : 40 course 4 : 40
course 5 : 30
Final Marks:
course 1 : 72 course 2 : 74 course 3 : 72 course 4 : 77
course 5 : 77

```

Code:

```
// File: CIE/Student.java
package CIE;

import java.util.Scanner;
public class Student {
    protected String usn;
    protected String name;
    protected int sem;

    // Method to input student details
    public void inputStudentDetails() {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter USN: ");
        this.usn = s.nextLine();
        System.out.print("Enter Name: ");
        this.name = s.nextLine();
        System.out.print("Enter Semester: ");
        this.sem = s.nextInt();
    }

    // Method to display student details
    public void displayStudentDetails() {
        System.out.println("USN: " + usn);
        System.out.println("Name: " + name);
        System.out.println("Semester: " + sem);
    }
}

// File: CIE/Internals.java
package CIE;
import java.util.Scanner;
public class Internals extends Student {
    protected int marks[] = new int[5];

    public void inputCIEmarks() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter Internal Marks for 5 courses: ");
        for (int i = 0; i < 5; i++) {
            System.out.print("Course " + (i + 1) + ": ");
            marks[i] = s.nextInt();
        }
    }
}
```

```

        }
    }

    // Method to display internal marks
    public void displayCIEmarks() {
        System.out.println("Internal Marks: ");
        for (int i = 0; i < 5; i++) {
            System.out.println("Course " + (i + 1) + ": " + marks[i]);
        }
    }
}

// File: SEE/Externals.java
package SEE;
import CIE.Internals; // Import Internals class from CIE package
import java.util.Scanner;

public class Externals extends Internals {
    protected int externalMarks[] = new int[5]; // Array to store external marks for
5 courses
    protected int finalMarks[] = new int[5]; // Array to store final marks (sum of
internal and external)

    // Constructor to initialize arrays
    public Externals() {
        externalMarks = new int[5];
        finalMarks = new int[5];
    }

    // Method to input external marks
    public void inputSEEmarks() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter External Marks for 5 courses: ");
        for (int i = 0; i < 5; i++) {
            System.out.print("Course " + (i + 1) + ": ");
            externalMarks[i] = s.nextInt();
        }
    }

    // Method to calculate final marks (sum of internal and external)
    public void calculateFinalMarks() {
        for (int i = 0; i < 5; i++) {
            finalMarks[i] = marks[i] + externalMarks[i]; // Internal + External
        }
    }
}

```

```

}

// Method to display final marks
public void displayFinalMarks() {
    displayStudentDetails(); // Display student details (from parent class)
    displayCIEmarks(); // Display internal marks (from parent class)

    System.out.println("External Marks: ");
    for (int i = 0; i < 5; i++) {
        System.out.println("Course " + (i + 1) + ": " + externalMarks[i]);
    }

    System.out.println("Final Marks: ");
    for (int i = 0; i < 5; i++) {
        System.out.println("Course " + (i + 1) + ": " + finalMarks[i]);
    }
}
}

File: Main.java
import SEE.Externals; // Import Externals class from SEE package
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter number of students: ");
        int n = sc.nextInt();

        // Array to store student data
        Externals[] students = new Externals[n];

        // Loop to input data for each student
        for (int i = 0; i < n; i++) {
            students[i] = new Externals();

            // Input student details, internal marks, and external marks
            students[i].inputStudentDetails();
            students[i].inputCIEmarks();
            students[i].inputSEEmarks();
            students[i].calculateFinalMarks();
        }

        // Display the final results of all students
    }
}

```

```

        for (int i = 0; i < n; i++) {
            students[i].displayFinalMarks();
            System.out.println("-----");
        }
    }
}

```

Output:

```

C:\Users\Admin\Documents\student>javac -d . CIE/Internals.java
C:\Users\Admin\Documents\student>javac -d . CIE/Student.java
C:\Users\Admin\Documents\student>javac -d . SEE/Externals.java
C:\Users\Admin\Documents\student>javac Main.java

C:\Users\Admin\Documents\student>java Main
Enter number of students: 1
Enter USN: 1286759
Enter Name: SIRI
Enter Semester: 3
Enter Internal Marks for 5 courses:
Course 1: 34
Course 2: 35
Course 3: 32
Course 4: 37
Course 5: 40
Enter External Marks for 5 courses:
Course 1: 38
Course 2: 39
Course 3: 40
Course 4: 40
Course 5: 30
USN: 1286759
Name: SIRI
Semester: 3
Internal Marks:
Course 1: 34
Course 2: 35
Course 3: 32
Course 4: 37
Course 5: 40
External Marks:
Course 1: 38
Course 2: 39
Course 3: 40
Course 4: 40
Course 5: 30
Final Marks:
Course 1: 72
Course 2: 74
Course 3: 72
Course 4: 77
Course 5: 70

```

Program 7

Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" and derived class called "Son" which extends the base class. In Father class, implement a constructor which takes the age and throws the exception Wrong Age() when the input age=father's age.

<p>SURYA Gold Date _____ Page _____</p> <p>7 Write a program that demonstrates handling of exception in inheritance tree. Create a base class called as "father" and a derived class son which extends the base class. In father's class implement a constructor which takes the age and throws the exception WrongAge() when the input age < 0. In son's class implement a constructor that uses both father and son's age and throws an exception if son's age ≥ father's age</p> <pre> import java.util.Scanner; class WrongAge extends Exception { public WrongAge (String message) { super (message); } } class SonAgeException extends Exception { public SonAgeException (String message) { super (message); } } class Father { private int age; public Father (int age) throws WrongAge { if (age < 0) throw new WrongAge ("Father's age cannot be negative"); this.age = age; } } </pre>	<p>SURYA Gold Date _____ Page _____</p> <pre> public int getAge () { return age; } class Son extends Father { private int sonAge; public Son (int fatherAge, int sonAge) throws WrongAge, SonAgeException, NegativeSonAgeException { super (fatherAge); if (sonAge < 0) throw new NegativeSonAgeException ("Son's age cannot be negative"); if (sonAge ≥ fatherAge) throw new SonAgeException ("Son's age cannot be greater than or equal to Father's age"); this.sonAge = sonAge; } public int getSonAge () { return sonAge; } } public class Fatherson { public static void main (String [] args) { Scanner sc = new Scanner (System.in); while (true) { </pre>
---	--

```

    (Date _____) (Page _____)
System.out.print("Enter father's age:");
int fatherAge = sc.nextInt();
System.out.print("Enter Son's age:");
int sonAge = sc.nextInt();
try {
    Son son = new Son(fatherAge, sonAge);
    System.out.println("Accepted Successfully");
} catch(WrongAge e) {
    System.out.println(e.getMessage());
} catch(SonAgeException e) {
    System.out.println(e.getMessage());
} catch(NegativeSonAgeException e) {
    System.out.println(e.getMessage());
}
System.out.println("Would you like to re-enter details (Y/N)?");
String input = sc.next();
if (input.equalsIgnoreCase("n")) {
    break;
}
sc.close();
}

Output:
Enter Father's Age: 40
Enter Son's Age: -10
Son's Age cannot be negative
Would you like to re-enter details (Y/N)?

```

SURYA Gold
(Date _____) (Page _____)

Enter Father's Age: 40
Enter Son's Age: 40
Son's Age cannot be greater than or equal to father's age.
Would you like to re-enter details (Y/n)?
Y

Enter Father's Age: -35
Enter Son's Age: 20
Father's Age cannot be negative.
Would you like to re-enter details (Y/n)?
Y

Enter Father's Age: 40
Enter Son's Age: 10
Accepted Successfully
Would you like to re-enter details (Y/n)?
N

```

import java.util.Scanner;

class WrongAge extends Exception {

    public WrongAge(String message) {
        super(message);
    }
}

```

```

class SonAgeException extends Exception {

    public SonAgeException(String message) {

        super(message);

    }
}

class NegativeSonAgeException extends Exception {

    public NegativeSonAgeException(String message) {

        super(message);
    }
}

class Father {

    private int age;

    public Father(int age) throws WrongAge {

        if (age < 0) {

            throw new WrongAge("Father's age cannot be negative");
        }

        this.age = age;
    }

    public int getAge() {

        return age;
    }
}

class Son extends Father {

    private int sonAge;

    public Son(int fatherAge, int sonAge) throws WrongAge, SonAgeException,
NegativeSonAgeException {

        super(fatherAge);

        if (sonAge < 0) {

            throw new NegativeSonAgeException("Son's age cannot be negative");
        } if (sonAge >= fatherAge) {
    }
}

```

```

        throw new SonAgeException("Son's age cannot be greater than or equal to father's
age");    }

    this.sonAge = sonAge;

} public int getSonAge() {

    return sonAge; }}

public class FatherSon {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        while (true) {

            System.out.print("Enter Father's Age: ");

            int fatherAge = sc.nextInt();

            System.out.print("Enter Son's Age: ");

            int sonAge = sc.nextInt();

            try { Son son = new Son(fatherAge, sonAge); // Create Son object

                System.out.println("Accepted Successfully");

            } catch (WrongAge e) {

                System.out.println(e.getMessage());

            } catch (SonAgeException e) {

                System.out.println(e.getMessage());

            } catch (NegativeSonAgeException e) {

                System.out.println(e.getMessage());

            }

            System.out.println("Would you like to re-enter details (Y/n)?");

            String input = sc.next();

            if (input.equalsIgnoreCase("n")) {

```

```
        break;  
  
    }  
  
    sc.close();  
}  
}
```

Output:

```
C:\Users\HP\Documents>javac FatherSon.java  
  
C:\Users\HP\Documents>java FatherSon  
Name:Archie Jain  
USN:1BM23CS049  
Enter Father's Age: 40  
Enter Son's Age: -10  
Son's age cannot be negative  
Would you like to re-enter details (Y/n)?  
Y  
Enter Father's Age: 40  
Enter Son's Age: 40  
Son's age cannot be greater than or equal to father's age  
Would you like to re-enter details (Y/n)?  
Y  
Enter Father's Age: 40  
Enter Son's Age: 10  
Accepted Successfully  
Would you like to re-enter details (Y/n)?  
Y  
Enter Father's Age: -35  
Enter Son's Age: 20  
Father's age cannot be negative  
Would you like to re-enter details (Y/n)?  
N
```

Program 8

Write a program which creates two threads, one thread displaying “BMS College of Engineering” once every ten seconds and another displaying “CSE” once every two seconds.

SURYA Gold
Date _____ Page _____

8. Write a program which creates two threads, one thread displaying “BMS College of engineering” once every ten seconds and another displaying “CSE” once every two seconds.

```
class BMSThread extends Thread {  
    private int count;  
    public BMSThread (int count) {  
        this.count = count;  
    }  
    public void run() {  
        for (int i=0; i<count; i++) {  
            System.out.println ("BMS college of Engineering");  
        }  
    }  
}  
  
class CSEThread extends Thread {  
    private int count;  
    public CSEThread (int count) {  
        this.count = count;  
    }  
    public void run() {  
        for (int i=0; i<count; i++) {  
            System.out.println ("CSE");  
        }  
    }  
}
```

```

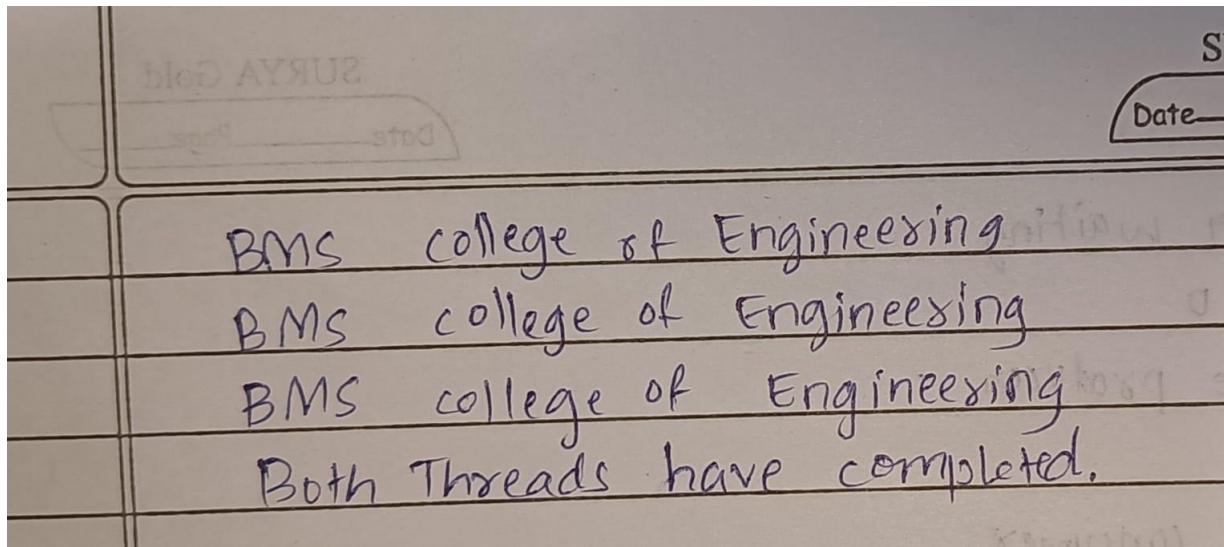
Date _____ Page _____
    Thread.sleep(2000);
} catch (InterruptedException e) {
    System.out.println(e);
}
}

public class Main {
    public static void main (String [] args) {
        int printCount = 5;
        BMSThread bmsThread = new BMSThread (printCount);
        CSEThread cseThread = new CSEThread (printCount);
        bmsThread.start();
        cseThread.start();
    }

    try {
        bmsThread.join();
        cseThread.join();
    } catch (InterruptedException e) {
        System.out.println(e);
    }
    System.out.println ("Both threads have completed");
}

Output
BMS college of Engineering
CSE
CSE
CSE
CSE
BMS college of engineering

```



```
class BMSThread extends Thread {  
    private int count;  
  
    public BMSThread(int count) {  
        this.count = count;  
    }  
    public void run() {  
        for (int i = 0; i < count; i++) {  
            System.out.println("BMS College of Engineering");  
            try {  
                Thread.sleep(10000); // Sleep for 10 seconds  
            } catch (InterruptedException e) {  
                System.out.println(e);  
            }  
        }  
    }  
}  
  
class CSEThread extends Thread {  
    private int count;  
  
    public CSEThread(int count) {  
        this.count = count;  
    }  
    public void run() {  
        for (int i = 0; i < count; i++) {  
            System.out.println("CSE");  
            try {  
                Thread.sleep(2000); // Sleep for 2 seconds  
            } catch (InterruptedException e) {  
                System.out.println(e);  
            }  
        }  
    }  
}
```

```

    }}}}}

public class Main {
    public static void main(String[] args) {
        int printCount = 5; // Number of times each message should be printed
        BMSThread bmsThread = new BMSThread(printCount);
        CSEThread cseThread = new CSEThread(printCount);
        bmsThread.start();
        cseThread.start();
    }

    try {
        bmsThread.join();
        cseThread.join();
    } catch (InterruptedException e) {
        System.out.println(e);
    }

    System.out.println("My name is ArchieJain.");
    System.out.println("My usn is 1BM23CS049.");
}
}

```

Output:

```

C:\foder>java Main
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
BMS College of Engineering
BMS College of Engineering
BMS College of Engineering
My name is ArchieJain.
My usn is 1BM23CS049.

```

Program 9

Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a Number Format Exception. If Num2 were Zero, the program would throw an Arithmetic Exception. Display the exception in a message dialog box.

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

class SwingDemo {
    SwingDemo() {
        JFrame jfrm = new JFrame("Divider App");
        jfrm.setSize(250, 150);
        jfrm.setLayout(new FlowLayout());
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JLabel jlab = new JLabel("Enter the divisor and dividend : ");

        JTextField ajtf = new JTextField(8);
        JTextField bjtf = new JTextField(8);

        JButton button = new JButton("Calculate");

        JLabel err = new JLabel();
        JLabel alab = new JLabel();
        JLabel blab = new JLabel();
        JLabel anslab = new JLabel();

        jfrm.add(err);
        jfrm.add(jlab);
        jfrm.add(ajtf);
        jfrm.add(bjtf);
        jfrm.add(button);
        jfrm.add(alab);
        jfrm.add(anslab);
    }
}

jfrm.add(blab);
jfrm.add(anslab);

ActionListener J = new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        System.out.println("Action event from a text field");
    }
};

ajtf.addActionListener(J);
bjtf.addActionListener(J);

button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        try {
            int a = Integer.parseInt(ajtf.getText());
            int b = Integer.parseInt(bjtf.getText());
            int ans = a/b;
            alab.setText("\nA=" + a);
            blab.setText("\nB=" + b);
            anslab.setText("\nAns=" + ans);
        } catch (NumberFormatException e) {
            alab.setText("");
            blab.setText("");
            anslab.setText("");
            err.setText("Enter only Integers!");
        } catch (ArithmaticException e) {
            alab.setText("");
            blab.setText("");
            anslab.setText("");
            err.setText("Enter only Integers!");
        }
    }
});
```

SURYA Gold
Date _____ Page _____

```

err.setText("B should be Non-zero!");
}
}
}
jfrm.setVisible(true);

public static void main (String args[])
{
    SwingUtilities.invokeLater(new Runnable()
    {
        public void run()
        {
            new SwingDemo();
        }
    });
}

```

→ Enter the divider and dividend:

10 2

$A=10 \quad B=2 \quad \text{Ans}=5$

Enter the divider and dividend:

10 0

B should be Non zero!

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
class SwingDemo{
    SwingDemo(){

        JFrame jfrm = new JFrame("Divider App");
        jfrm.setSize(275, 150);
        jfrm.setLayout(new FlowLayout());
        // to terminate on close
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        // text label
        JLabel jlab = new JLabel("Enter the divider and dividend:");
        // add text field for both numbers
        JTextField ajtf = new JTextField(8);
        JTextField bjtf = new JTextField(8);
        // calc button
        JButton button = new JButton("Calculate");
        // labels
        JLabel err = new JLabel();
        JLabel alab = new JLabel();
        JLabel blab = new JLabel();
        JLabel anslab = new JLabel();
        // add in order :
    }
}

```

```

jfrm.add(err); // to display error bois
jfrm.add(jlab);
jfrm.add(ajtf);
jfrm.add(bjtf);
jfrm.add(button);
jfrm.add(alab);
jfrm.add(blab);
jfrm.add(anslab);

ActionListener l = new ActionListener() {
    public void actionPerformed(ActionEvent evt) { System.out.println("Action event from a text
field"); }
};

ajtf.addActionListener(l);
bjtf.addActionListener(l);
button.addActionListener(new ActionListener() { public void actionPerformed(ActionEvent evt) {
try{
    int a = Integer.parseInt(ajtf.getText()); int b = Integer.parseInt(bjtf.getText()); int ans =
a/b;

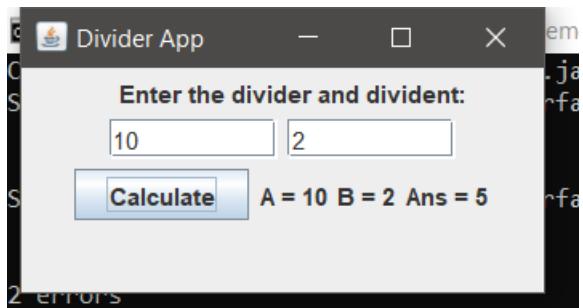
    alab.setText("\nA = " + a);
    blab.setText("\nB = " + b);
    anslab.setText("\nAns = " + ans);
}
catch(NumberFormatException e){
    alab.setText("");
    blab.setText("");
    anslab.setText("");
    err.setText("Enter Only Integers!"); }
catch(ArithmException e){ alab.setText("");
    blab.setText("");
    anslab.setText("");
    err.setText("B should be NON zero!"); }
}
});;
// display frame
jfrm.setVisible(true);
}

public static void main(String args[]){
    SwingUtilities.invokeLater(new Runnable()
{ public void run(){
    new SwingDemo();
}
}
}

```

```
); }}
```

Output:



Program 10

Demonstrate Inter process Communication and deadlock

The handwritten notes show two pages of code for a producer-consumer problem.

Page 1 (Left):

- Ques: Implementation of producer and consumer
- Code:

```
class Q {
    int n;
    boolean valueset = false;
    synchronized int get() {
        while (!valueset)
            try {
                System.out.println("\n Consumer waiting");
                wait();
            } catch (InterruptedException e) {
                System.out.println(" InterruptedException caught");
            }
        System.out.println(" Got: " + n);
        valueset = false;
        System.out.println("\n Intimate producer");
        notify();
        return n;
    }
    synchronized void put(int n) {
        while (valueset)
            try {
                System.out.println("\n Producer waiting");
                wait();
            } catch (InterruptedException e) {
                System.out.println(" InterruptedException caught");
            }
        this.n = n;
        valueset = true;
        System.out.println(" put: " + n);
        System.out.println("\n Intimate consumer");
    }
}
```

Page 2 (Right):

- Notify();
- class producer implements Runnable
- Q q;
- Producer(Q q) {
 this.q = q;
 new Thread(this, "Producer").start();
}
- public void run() {
 int i=0;
 while (i<5) {
 q.put(i+1);
 }
 class consumer implements Runnable
 Q q;
 Consumer(Q q) {
 this.q = q;
 new Thread(this, "consumer").start();
 }
 public void run() {
 int i=0;
 while (i<5) {
 int x=q.get();
 System.out.println(" consumed: " + x);
 i++;
 }
 }
 class pifixed
 public static void main (String args[]) {
 Q q = new Q();
 new producer(q);
 new consumer(q);
 System.out.println("Press: Control-C to stop.");
 }
}

~~Interprocess Communication~~

Output

Press Control-C to stop

put : 0

Intimate consumer

producer

producer waiting

got : 0

Intimate producer

put = 1

Intimate consumer

producer waiting

consumed : 0

Got : 1

Intimate producer

Consumed : 1

put : 2

Intimate consumer

producer waiting

Got : 2

Intimate producer

Consumed : 2

Put : 3

Intimate consumer

producer waiting

Got : 3

Intimate producer

Consumed : 3

put : 4

Intimate consumer

producer Got : 4

Intimate producer

Consumed : 4

Get

SURYA Gold
Date _____ Page _____

Dead lock

```

class A {
    synchronized void foo(B b) {
        String name = Thread.currentThread().getName();
        System.out.println(name + " entered A.foo");
        try {
            Thread.sleep(1000);
        } catch (Exception e) {
            System.out.println("A interrupted");
        }
        System.out.println(name + " trying to call B.last()");
        b.last();
    }

    void last() {
        System.out.println("Inside A.last");
    }
}

class B {
    synchronized void bar(A a) {
        String name = Thread.currentThread().getName();
        System.out.println(name + " entered B.bar");
        try {
            Thread.sleep(1000);
        } catch (Exception e) {
            System.out.println("B interrupted");
        }
        System.out.println(name + " trying to call A.last()");
        a.last();
    }
}

```

Date _____ Page _____

```

void last() {
    System.out.println("Inside A.last");
}

class Deadlock implements Runnable {
    A a = new A();
    B b = new B();
    Deadlock() {
        Thread.currentThread().setName("MainThread");
        Thread t = new Thread(this, "RacingThread");
        t.start();
        a.foo(b);
    }
    public void run() {
        b.bar(a);
    }
    public static void main(String args[]) {
        new Deadlock();
    }
}

MainThread entered A.foo
Racing Thread entered B.bar
MainThread trying to call B.last()
Inside A.last
Back in main thread
Racing Thread trying to call A.last()
Inside A.last
Back in other thread.

```

```

class Q {
    int n;
    boolean valueSet = false;
    synchronized int get() {

```

```

while(!valueSet)
try {
System.out.println("\nConsumer waiting\n");
wait();
} catch(InterruptedException e) {
System.out.println("InterruptedException caught");
}
System.out.println("Got: " + n);
valueSet = false;
System.out.println("\nIntimate Producer\n");
notify();
return n;
}
synchronized void put(int n) {
while(valueSet)
try {
System.out.println("\nProducer waiting\n");
wait();
} catch(InterruptedException e) {
System.out.println("InterruptedException caught");
}
this.n = n;
valueSet = true;
System.out.println("Put: " + n);
System.out.println("\nIntimate Consumer\n");
notify();
}
}
class Producer implements Runnable {
Q q;
Producer(Q q) {
this.q = q;
new Thread(this, "Producer").start();
}
public void run() {
int i = 0;
while(i<15) {
q.put(i++);
}
}
}
class Consumer implements Runnable {
Q q;
Consumer(Q q) {

```

```
this.q = q;
new Thread(this, "Consumer").start();
}
public void run() {
    int i=0;
    while(i<15) {
        int r=q.get();
        System.out.println("consumed:"+r);
        i++;
    }
}
class PCFixed {
    public static void main(String args[]) {
        Q q = new Q();
        new Producer(q);
        new Consumer(q);
        System.out.println("Press Control-C to stop.");
    }
}
```

```

Press Control-C to stop.
Put: 0
Intimate Consumer
Producer waiting
Got: 0
Intimate Producer
Put: 1
consumed:0
Intimate Consumer
Producer waiting
Got: 1
Intimate Producer
consumed:1
Put: 2
Intimate Consumer
Producer waiting
Got: 2
Intimate Producer
consumed:2
Put: 3
Intimate Consumer
Producer waiting
Got: 3
Intimate Producer
consumed:3
Put: 4
Intimate Consumer
Got: 4
Intimate Producer
consumed:4
Put: 5
Intimate Consumer
Producer waiting
Got: 5
Intimate Producer
consumed:5

```

```

class A {
    synchronized void foo(B b) {
        String name = Thread.currentThread().getName();
        System.out.println(name + " entered A.foo");
        try {
            Thread.sleep(1000);
        } catch(Exception e) {
            System.out.println("A Interrupted");
        }
        System.out.println(name + " trying to call B.last()");
        b.last();
    }
    void last() {
        System.out.println("Inside A.last");
    }
}

```

```
class B {  
    synchronized void bar(A a) {  
        String name = Thread.currentThread().getName();  
        System.out.println(name + " entered B.bar");  
        try {  
            Thread.sleep(1000);  
        } catch(Exception e) {  
            System.out.println("B Interrupted");  
        }  
        System.out.println(name + " trying to call A.last()");  
        a.last();  
    }  
    void last() {  
        System.out.println("Inside A.last");  
    }  
}  
  
class Deadlock implements Runnable {  
    A a = new A();  
    B b = new B();  
    Deadlock() {  
        Thread.currentThread().setName("MainThread");  
        Thread t = new Thread(this, "RacingThread");  
        t.start();  
        a.foo(b); // get lock on a in this thread.  
        System.out.println("Back in main thread");  
    }  
    public void run() {  
        b.bar(a); // get lock on b in other thread.  
        System.out.println("Back in other thread");  
    }  
    public static void main(String args[]) {  
        new Deadlock();  
    }  
}
```

```
C:\Users\HP\Documents>javac Deadlock.java  
C:\Users\HP\Documents>java Deadlock  
MainThread entered A.foo  
RacingThread entered B.bar  
MainThread trying to call B.last()  
RacingThread trying to call A.last()  
Inside A.last  
Back in main thread  
Inside A.last  
Back in other thread
```